6.456: Adaptive Array Processing
Fall 2021
Lab 2: Signal processing for dummies

The objective of this lab is to evaluate your practical signal processing skills, and to remind (or teach?) you the basic stuff you'll need for the rest of the class. For this specific lab, you will be required to **provide a preliminary lab report at the end of the lab**. I do not need anything fancy or well written here, I just want to know how much you did in the time allotted for the lab. Your final lab report is due on October 7. You can either **work alone or in pairs**.

BEFORE THE LAB:

- record yourself using a computer and a microphone. Just record a few words, e.g. "I love array processing"
- also record yourself whistling a relatively long continuous tones (a few seconds is enough). If you do not like to whistle, you can record any tonal signal (the tone of an old phone is perfect)

Make sure to produce 16 bits .wav files with a sampling frequency fs=44 kHz. You can find online tutorials to do so, any free audio software can be used.

THINGS TO BRING TO THE LAB

- a laptop with Matlab
- headphones

THINGS TO REMEMBER DURING THE LAB

- all figures must have a title and labels on the x and y axes
- whenever you do a figure with Matlab, always specify the axes
    - use *plot(X,Y)* to plot Y versus X. Do not use *plot(Y).*
    - use *imagesc(x,y,C)* to plot the matrix (i.e. image) C versus x and y. Do not use *plot(C)*.
- most physical quantities have units
- make sure to look at exercise 5 during the lab
- ask any signal processing question you have, even if it's not covered by the lab

The lab description start on next page.

1. Time domain, frequency domain, time-frequency domain

Open your speech sound file with Matlab.

1.1 Plot the signal in the time domain. Make sure to correctly define the time axis

1.2 Plot the signal in the frequency domain. Use the *fft* function. Make sure to correctly define the frequency axis.

1.3 Plot the signal in the time-frequency domain. Use the *spectrogram* function. Choose the spectrogram parameters (window, noverlap, nfft) to obtain nice picture. Make sure to correctly define the time and frequency axis (they can be obtained as output of the *spectrogram* function).

2. Sampling frequency and time aliasing

Open your whistle sound file with Matlab. Play the sound with Matlab (use the *audioplayer* function) – use your headphone.

2.1 Identify the main frequency of the tone.

2.2 Trick Matlab to play the sound with a wrong sampling frequency (try fs*2 and fs/2). Explain what you hear.

2.3 What is the minimal sampling frequency fmin if we want to decimate the signal?

2.4 We will now decimate the signal. We want the sampling frequency after decimation to be as small as possible, while being greater than fmin. We will try two decimation methods.

Let's call your original signal s (with sampling frequency fs) and the signal after decimation s_new (with sampling frequency fs_new). To decimate s by a factor of N, try the two lines of code

- *s_new=s(1:N:end)*
- *s_new=decimate(s, N)*

In both case, fs_new=fs/N. For each method, plot the signal in the time domain, in the frequency domain, and in the time-frequency domain. Listen to the decimated signals. Explain everything

2.5 Redo question 2.4, but decimate so that fs_new≈fmin/2. Carefully explain the difference between the two decimation methods.

3. Frequency resolution and interpolation in the frequency domain

3.1 Generate a sin wave of frequency f0=7 Hz and of length equal to 2 periods. Use a sampling frequency fs=25 Hz. Plot the signal in the time domain and in the frequency domain.

3.2 Redo question 3.1, but increase the sampling frequency (fs=250 Hz). Explain the result.

3.3 Redo question 3.1, but increase the length to 10 periods. Explain the results

3.4 Redo question 3.1, but add trailing zeros at the end of the sin wave, so that the overall signal has the same length than in 3.3 (this is called "zero-padding"). Explain the results

3.5 In fact, there is no need to create a signal with trailing zeros to perform zero-padding with Matlab. Use the signal generated in 3.1 and obtain the results of 3.4 using the zero padding option from the *fft* function.

4. Interpolation in the time domain

For this exercise, re-use the signal from question 3.1. Make sure that its length N is an odd number (remove 1 sample if necessary). The objective is to interpolate this (time-domain) signal.

4.1 To interpolate in the time-domain, one can use zero padding in the frequency domain. Let us interpolate the signal by a factor Ninterp (i.e. Ninterp=10). To do so :

- compute the Fourier transform of the signal (use *fft* with nfft=N)
- add many zeros IN THE MIDDLE of the frequency domain signal to obtain a vector of length N*Ninterp
- compute the inverse Fourier transform (use *ifft* with nfft=N)

Plot the original signal and the interpolated signal on a single figure

4.2 Explain the similarity/difference between zero padding in the time domain and zero padding in the frequency domain. For zero padding in the frequency domain, explain why the zeros must be added in the middle of the signal and not at the end.

5. Power Spectrum and Power Spectral Density

5.1 Simulate a sin wave s of frequency f0=100.33 Hz and length T=30 sec, sampled at fs=1000 Hz. Compute its power spectral density and power spectrum using

```
N=length(s);
nfft=N;
[psd_s,f_per]=periodogram(s,rectwin(N),nfft,fs);
[ps_s,f_per]=periodogram(s,rectwin(N),nfft,fs,'power');
```

Explain the code (in particular the input parameter of the *periodogram* function) and plot the results.

5.2 Redo question 5.1, but with T=1 sec. Plot the results on top of the previous figure, and explain the results

5.3 Mathematically derive the power of s. Numerically compute the power of s using

- the time domain definition
- a periodogram plot

5.3 Create a noisy signal x=s+n, with s from question 5.1 and n a normally distributed random noise (use the function *randn*). Here, we can pretend that s is an unknown signal of interest, and that x is a noisy measurement. Estimate the frequency of s and its power using x.

5.4 Redo question 5.3, but with T=1 sec.

5.5 OPTIONAL QUESTION: Redo question 5.1 using the *fft* function (do not use the *periodogram* function). Read https://www.mathworks.com/help/signal/ug/power-spectral-density-estimates-using-fft.html for instructions.